

A Discrete Time Markov Chain Model for High Throughput Bidirectional Fano Decoders

Ran Xu*, Graeme Woodward[†], Kevin Morris* and Taskin Kocak*

*Centre for Communications Research, Department of Electrical and Electronic Engineering
University of Bristol, Bristol, UK

[†]Telecommunications Research Laboratory (TRL), Toshiba Research Europe Limited, 32 Queen Square, Bristol, UK

Abstract—The bidirectional Fano algorithm (BFA) can achieve at least two times decoding throughput compared to the conventional unidirectional Fano algorithm (UFA). In this paper, bidirectional Fano decoding is examined from the queuing theory perspective. A Discrete Time Markov Chain (DTMC) is employed to model the BFA decoder with a finite input buffer. The relationship between the input data rate, the input buffer size and the clock speed of the BFA decoder is established. The DTMC based modelling can be used in designing a high throughput parallel BFA decoding system. It is shown that there is a trade-off between the number of BFA decoders and the input buffer size, and an optimal input buffer size can be chosen to minimize the hardware complexity for a target decoding throughput in designing a high throughput parallel BFA decoding system.

Index Terms—Bidirectional Fano algorithm, high throughput decoding, queuing theory, sequential decoding.

I. INTRODUCTION

Sequential decoding is one method for decoding convolutional codes [1]. Compared to the well-known Viterbi algorithm, the computational effort of sequential decoding is adaptive to the signal-to-noise-ratio (SNR). When the SNR is relatively high, the computational complexity of sequential decoding is much lower than that of Viterbi decoding. Additionally, sequential decoding can decode very long constraint length convolutional codes since its computational effort is independent of the constraint length. Thus, a long constraint length convolutional code can be used to achieve a better error rate performance. There are mainly two types of sequential decoding algorithms which are known as the Stack algorithm [2] and the Fano algorithm [3]. The Fano algorithm is more suitable for hardware implementations since it does not require extensive sorting operations or large memory as the Stack algorithm [4][5].

High throughput decoding is of research interest due to the increasing data rate requirement. The baseband signal processing is becoming more and more power and area hungry. For example, to achieve the required high throughput, the WirelessHD specification proposes simultaneous transmission of eight interleaved codewords, each encoded by a convolutional code [6]. It is straightforward to use eight parallel Viterbi decoders to achieve multi-Gbps decoding throughput. Since sequential decoding has the advantage of lower hardware complexity and lower power consumption compared to Viterbi

decoding [4][5], we are motivated to consider the usage of sequential decoding in high throughput applications when the SNR is relatively high. In a practical implementation of a sequential decoder, an input buffer is required due to the variable computational effort of each codeword. The contribution of this work is that the bidirectional Fano decoder with an input buffer was modelled by a Discrete Time Markov Chain (DTMC) and the relationship between the input data rate, the input buffer size and the clock speed of the BFA decoder was established. The trade-off between the number of BFA decoders and the input buffer size in designing a high throughput parallel BFA decoding system was also presented.

The rest of the paper is organized as follows. In Section II, the bidirectional Fano algorithm is reviewed and the system model is given. The BFA decoder with an input buffer is analyzed by queuing theory in Section III, and the simulation results are presented in Section IV. Section V is about choosing the optimal input buffer size in designing a parallel BFA decoding system, and the conclusions are drawn in Section VI.

II. SYSTEM MODEL FOR BFA DECODER

A. Bidirectional Fano Algorithm

In the conventional unidirectional Fano algorithm (UFA), the decoder starts decoding from state zero. During each iteration of the algorithm, the current state may move forward, move backward, or stay at the current state. The decision is made based on the comparison between the threshold value and the path metric. If a forward movement is made, the threshold value needs to be tightened. If the current state cannot move forward or backward, the threshold value needs to be loosened. A detailed flowchart of the Fano algorithm can be found in [1]. In [7], a bidirectional Fano algorithm (BFA) was proposed, in which there is a forward decoder (FD) and a backward decoder (BD) working in parallel. Both the FD and the BD decode the same codeword from the start state and the end state in the opposite direction simultaneously. The decoding will terminate if the FD and the BD merge with each other or reach the other end of the code tree. Compared to the conventional UFA, the BFA can achieve a much higher decoding throughput due to the reduction in computational effort and the parallel processing of the two decoders. A detailed discussion on the BFA can be found in [7].

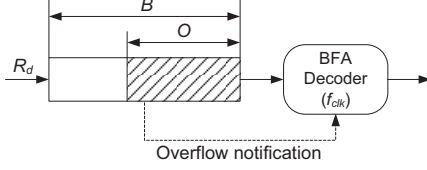


Fig. 1. System model for BFA decoder with overflow notification from the input buffer

B. System Model

Since the computational effort of sequential decoding is variable, an input buffer is used to accommodate the codewords to be decoded. The system model for a BFA decoder with an input buffer is shown in Fig. 1. It is assumed that there is continuous data stream input to the buffer whose raw data rate is R_d bps. The length of the input buffer is B , which means that it can accommodate up to B codewords, in addition to the one the decoder works on. The clock frequency of the BFA decoder is f_{clk} Hz and it is assumed that the BFA decoder can execute one iteration per clock cycle. In the BFA decoding, the number of clock cycles to decode one codeword follows the Pareto distribution, and the Pareto exponent is a function of the SNR and the code rate. A higher SNR or a lower code rate results in a higher Pareto exponent [7]. As shown in Fig. 1, there is an overflow notification from the input buffer to the BFA decoder. The occupancy of the input buffer is observed and the currently decoded codeword will be erased if the input buffer gets full. As a result, the total number of codewords consists of the following:

$$N_{total} = N_{decoded} + N_{erased}. \quad (1)$$

In order to evaluate the performance of a BFA decoder affected by the introduced parameters such as R_d , f_{clk} and B , a metric called failure probability (P_f) is defined as follows:

$$P_f = \frac{N_{erased}}{N_{total}} = \frac{N_{erased}}{N_{decoded} + N_{erased}}, \quad (2)$$

where P_f is similar to the frame error rate (P_F) which is caused by the decoding errors. The total frame error rate is:

$$P_t = P_f + P_F. \quad (3)$$

In designing the system, R_d , f_{clk} and B need to be chosen properly to ensure that:

$$P_t \approx P_F. \quad (4)$$

In this paper, $P_f = 0.01 \times P_F$ is adopted as the target failure probability (P_{target}). How to choose R_d , f_{clk} and B to make a BFA decoder achieve P_{target} will be discussed next.

III. DTMC BASED MODELLING ON BFA DECODER

The effect of the input buffer has been investigated for iterative decoders such as Turbo decoder [8] and LDPC decoder [9]-[11]. The non-deterministic decoding time nature of the BFA is similar to that of Turbo decoding and LDPC

decoding. A modelling strategy similar to that introduced in [11] is used to analyze the BFA decoder with input buffer.

The relationship between the input data rate (R_d), the input buffer size (B) and the clock speed of the decoder (f_{clk}) can be found via simulation. Another way to analyze the system is to model it based on queuing theory. The BFA decoder with an input buffer can be treated as a **D/G/1/B** queue, in which **D** means that the input data rate is deterministic, **G** means that the decoding time is generic, **1** means that there is one decoder and **B** is the number of codewords the input buffer can hold. The state of the BFA decoder is represented by the input buffer occupancy (O) when a codeword is decoded, which is measured in terms of branches stored in the buffer. $O(n)$ and $O(n+1)$ have the following relationship:

$$O(n+1) = O(n) + [T_s(n) \cdot R_d - L_f], \quad (5)$$

where $O(n+1)$ is the input buffer occupancy when the n^{th} codeword is decoded, $T_s(n)$ is the decoding time of the n^{th} codeword by the BFA decoder and L_f is the length of a codeword in terms of branches. $[x]$ denotes the operation to get the nearest integer of x . The speed factor of the BFA decoder is defined as the ratio between f_{clk} and R_d [1]:

$$\mu = \frac{f_{clk}}{R_d}. \quad (6)$$

If f_{clk} is normalized to 1, Eq. (5) can be changed to:

$$O(n+1) = O(n) + \left[\frac{T_s(n)}{\mu} - L_f \right]. \quad (7)$$

The state of the input buffer at time $n+1$ is only decided by the state at time n and the decoding time $T_s(n)$. At the same time, $T_s(n)$ and $T_s(n+1)$ are *i.i.d.*. As a result, the state of the input buffer is a Discrete Time Markov Chain (DTMC). $T_s(n)$ follows the Pareto distribution for the BFA decoding and is in the unit of *clock cycle/codeword*. The following equation can be used to describe the Pareto distribution:

$$Prob(T_s > T) \approx A \cdot \left(\frac{T}{T_{min}} \right)^{-\beta}, \quad (8)$$

where T_{min} is the minimum decoding time which is L_f clock cycles in the considered model. The Pareto exponent β is a function of the SNR and the code rate. Fig. 2 shows the simulated and approximated (based on Eq. (8)) Pareto distributions for both the UFA and the BFA at $E_b/N_0=4\text{dB}$ and 5dB . It can be seen that as the SNR increases, the Pareto exponent increases, and for the same SNR the BFA has a higher Pareto exponent compared to the UFA. The simulated Pareto distribution of T_s , which is more accurate compared to the approximated distribution based on Eq. (8), will be used in the following analysis. The difference between $O(n+1)$ and $O(n)$ is defined as:

$$\Delta(n) = O(n+1) - O(n) = \left[\frac{T_s(n)}{\mu} - L_f \right]. \quad (9)$$

Fig. 3 shows that the total number of states of the input buffer with size B is:

$$\Omega = B \cdot L_f. \quad (10)$$

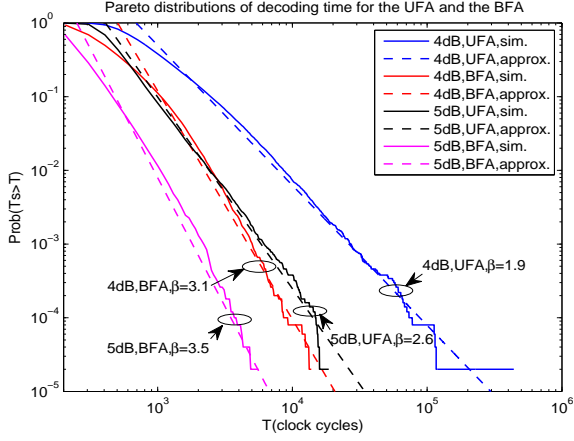


Fig. 2. Simulated and approximated Pareto distributions for the UFA and the BFA at $E_b/N_0=4\text{dB}$ and 5dB . The code rate is $R=1/3$.

The state transition diagram is shown in Fig. 4. As a result, the state transition probability matrix of the input buffer is:

$$P_T = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1\Omega} \\ P_{21} & P_{22} & \cdots & P_{2\Omega} \\ \vdots & \vdots & \ddots & \vdots \\ P_{\Omega 1} & P_{\Omega 2} & \cdots & P_{\Omega \Omega} \end{pmatrix}, \quad (11)$$

where P_{ij} is the state transition probability from S_i to S_j , which can be calculated as follows:

$$P_{ij} = \begin{cases} \sum_{k=\Delta_{\min}}^{-(i-1)} p_{\Delta+k}, & j = 1 \\ p_{\Delta+(j-i)}, & 1 < j < \Omega \\ 1 - \sum_{k=1}^{\Omega-1} P_{ik}, & j = \Omega \end{cases}, \quad (12)$$

where $p_{\Delta+w} = \text{Prob}(\Delta = w)$ and $\Delta_{\min} = \lceil \frac{\min(T_s)}{\mu} - L_f \rceil$. The value of $p_{\Delta+w}$ can be estimated from the simulated distribution of T_s as shown in Fig. 2. The initial state probability ($n=0$) of the input buffer is:

$$\pi(0) = (\pi_1(0), \pi_2(0), \dots, \pi_{\Omega}(0)) = (1, 0, \dots, 0). \quad (13)$$

The steady state probability of the input buffer is then:

$$\Pi = \lim_{n \rightarrow \infty} \pi(n) = \lim_{n \rightarrow \infty} \pi(0) \cdot P_T^n. \quad (14)$$

The failure probability of the decoder can be calculated by:

$$P_f = \sum_{i=1}^{\Omega} \Pi(i) \cdot p_{\Delta_{\Omega-i}}^+, \quad (15)$$

where $p_{\Delta_{\Omega-i}}^+ = \text{Prob}(\Delta > \Omega - i)$. The mean buffer occupancy can be calculated by:

$$O_{\text{mean}} = \frac{\sum_{i=1}^B i \cdot \sum_{j=1}^{L_f} \Pi((i-1) \cdot L_f + j)}{B} \times 100\%. \quad (16)$$

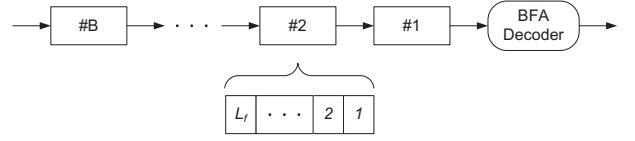


Fig. 3. BFA decoder with finite input buffer

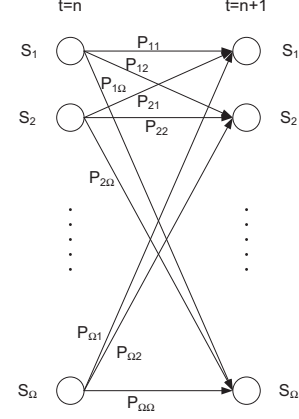


Fig. 4. Illustration of state transition

IV. SIMULATION RESULTS

Firstly, the semi-analytical results calculated by Eq.(15) are compared with the simulation results to validate the DTMC based modelling. The simulation setup is shown in Table 1. $E_b/N_0=4\text{dB}$ was used as an example, at which $P_{\text{target}} \approx 10^{-3}$. The convolutional code in the simulation was the one used in the WirelessHD specification [6]. The input buffer size B in the simulation takes the buffer within the BFA decoder into account. It can be seen from Fig. 5 that the semi-analytical results are quite close to the simulation results for both the UFA decoder and the BFA decoder, which means that the DTMC based modelling is accurate. For the input buffer size of $B=10$, the working speed factors of the UFA decoder and the BFA decoder are about $\mu=14$ and $\mu=3.6$, respectively. There is about 290% decoding throughput improvement by using the BFA decoder compared to the UFA decoder. If the input buffer size increases to $B=25$, the working speed factors will become about $\mu=8.7$ and $\mu=2.9$, respectively, resulting in about 200% decoding throughput improvement. As long as the distribution of T_s is known, P_f can be easily obtained for different values of speed factor and input buffer size. Simulation time can be greatly saved if the target P_f is very low (at high SNR) by using the DTMC based modelling. How to use the DTMC based modelling in designing a high throughput parallel BFA decoding system will be shown in the next section.

The input buffer occupancy distribution for the BFA decoder with $B=10$ at different speed factors is shown in Fig. 6, which was obtained from Eq.(14). The mean buffer occupancy in percentage calculated by Eq.(16) is shown in Fig. 7. For both $B=10$ and $B=25$ whose working speed factors are about 3.6

TABLE I
SIMULATION SETUP

| | |
|---|---|
| Code rate (R) | 1/3 |
| Generator polynomials | $g_0 = 133_8, g_1 = 171_8, g_2 = 165_8$ |
| Constraint length (K) | 7 |
| Branch metric calculation | 1-bit hard decision with Fano metric |
| Threshold adjustment value (δ) | 2 |
| Modulation | BPSK |
| Channel | AWGN |
| Information length (L) | 200 bits |
| Codeword length (L_f) | $L + K - 1 = 206$ branches |

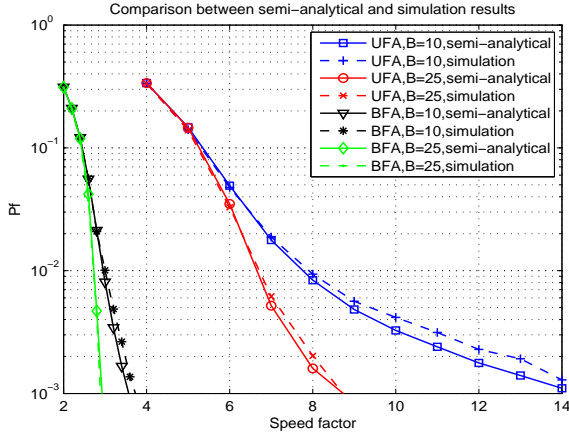


Fig. 5. Comparison between semi-analytical and simulation results (P_f vs μ) for UFA and BFA at $E_b/N_0=4$ dB

and 2.9, the mean buffer occupancies are about 17% and 25%, respectively. The decoding delay for $B=25$ is slightly higher than that for $B=10$, while the decoding throughput for $B=25$ is higher than that for $B=10$ as shown in Fig. 5.

V. INPUT BUFFER SIZE IN PARALLEL BFA DECODING

Unlike the Viterbi decoder, it is difficult to use pipelining in designing a high throughput BFA decoder due to the irregular decoding operations and the variable computational effort. Parallel processing is a promising strategy to achieve high throughput BFA decoding at multi-Gbps level. In order to achieve a specific decoding throughput, a number of BFA decoders ($N_{decoder}$) may need to be paralleled (as shown in Fig. 8) if a single BFA decoder cannot achieve the target average decoding throughput:

$$T_{target} = N_{decoder} \cdot R_d(B), \quad (17)$$

where R_d is a function of the input buffer size B . The total area of the parallel BFA decoders is:

$$\begin{aligned} \mathcal{A}_{total} &= \mathcal{A}_{decoder} + \mathcal{A}_{buffer} \\ &= N_{decoder} \cdot \mathcal{A}_{BFA} + N_{decoder} \cdot B \cdot \mathcal{A}_B. \end{aligned} \quad (18)$$

If the area ratio between a BFA decoder (\mathcal{A}_{BFA}) and an input buffer which can hold one codeword (\mathcal{A}_B) is $\alpha = \mathcal{A}_{BFA}/\mathcal{A}_B$, Eq.(17) will become:

$$T_{target} = \frac{\mathcal{A}_{total}}{\mathcal{A}_{BFA} + B \cdot \mathcal{A}_B} \cdot R_d(B) = \frac{\mathcal{A}_{total}}{\mathcal{A}_B} \cdot \frac{R_d(B)}{\alpha + B}. \quad (19)$$

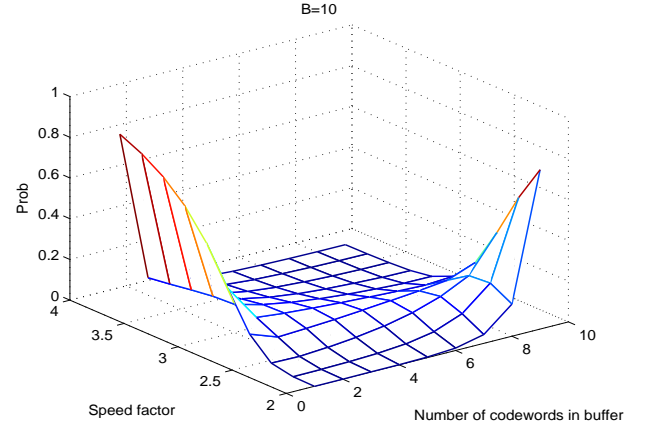


Fig. 6. Buffer occupancy distribution for BFA decoder at $E_b/N_0=4$ dB when $B=10$

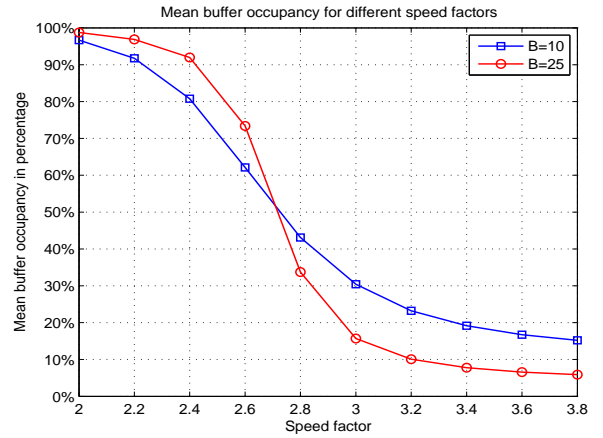


Fig. 7. Mean buffer occupancy for BFA decoder at $E_b/N_0=4$ dB when $B=10$ and $B=25$

It can be seen from Eq.(19) that for a fixed \mathcal{A}_{total} and \mathcal{A}_B , the decoding throughput of parallel BFA decoders changes with respect to the input buffer size B . The relationship between the input data rate R_d and input buffer size B is shown in Fig.9 which was obtained by the DTMC based modelling introduced in Section III. The clock speed of the BFA decoder is assumed to be $f_{clk}=1$ GHz. The normalized throughput with respect to the maximum throughput for different α values is shown in Fig.10. The value of α depends on the technology used in hardware implementation. It can be seen from Fig. 10 that there is an optimal choice of the input buffer size B to maximize the decoding throughput for a fixed area constraint. For example if $\alpha=16$, the optimal choice of the input buffer size will be 10. Equivalently, in order to achieve a target decoding throughput, the optimal choice of the input buffer size can minimize the hardware area, which will be explained by the following example.

■ Example

If the target decoding throughput is $T_{target}=1$ Gbps and two

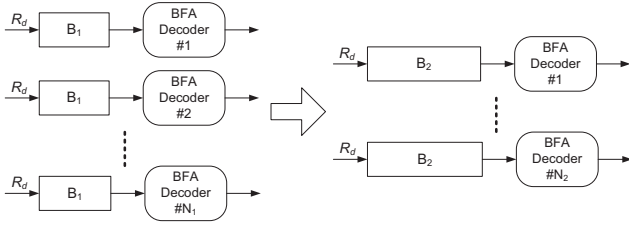


Fig. 8. Number of decoders vs input buffer size in parallel BFA decoding

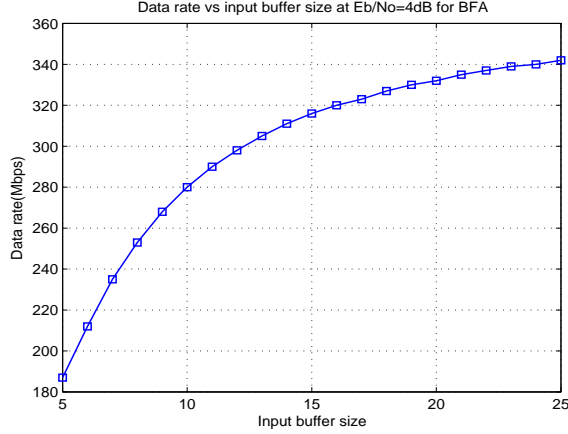


Fig. 9. Data rate vs input buffer size for BFA at $E_b/N_0=4\text{dB}$

input buffer sizes $B_1=5$ and $B_2=10$ are used, according to Eq.(17) and Fig.9, the number of parallel BFA decoders required are:

$$N_1 = 6 \text{ and } N_2 = 4. \quad (20)$$

When $B_1=5$ is used, the total area of the parallel BFA decoders will be:

$$\mathcal{A}_1 = N_1 \cdot \mathcal{A}_{BFA} + N_1 \cdot B_1 \cdot \mathcal{A}_B. \quad (21)$$

When $B_2=10$ is used, the total area of the parallel BFA decoders will be:

$$\mathcal{A}_2 = N_2 \cdot \mathcal{A}_{BFA} + N_2 \cdot B_2 \cdot \mathcal{A}_B. \quad (22)$$

If $\alpha=16$, the area reduction by using $B_2=10$ compared to $B_1=5$ will be:

$$\begin{aligned} \eta &= \left(\frac{\mathcal{A}_1}{\mathcal{A}_2} - 1 \right) \times 100\% \\ &= \left(\frac{N_1}{N_2} \cdot \frac{\alpha + B_1}{\alpha + B_2} - 1 \right) \times 100\% \approx 20\%. \end{aligned} \quad (23)$$

VI. CONCLUSION

In this paper, BFA decoder with input buffer was analyzed from the queuing theory perspective. The decoding system was modelled by a Discrete Time Markov Chain and the relationship between the input data rate, the input buffer size and the clock speed of the decoder was established. The working speed factor of the BFA decoder at each SNR can be easily found by the DTMC based modelling. The DTMC based modelling can be used in designing a high throughput parallel

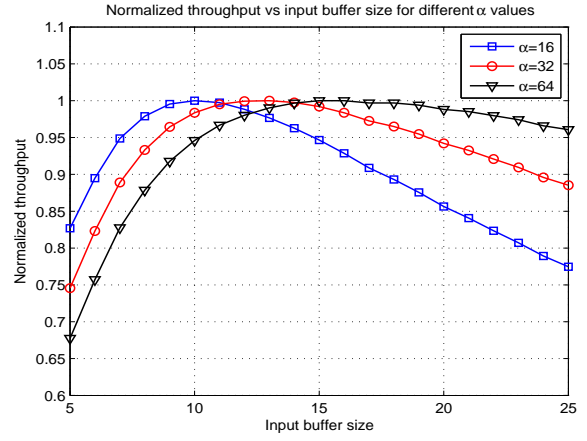


Fig. 10. Normalized throughput vs input buffer size for different α values at $E_b/N_0=4\text{dB}$

BFA decoding system. The trade-off between the number of BFA decoders and the input buffer size in designing a high throughput parallel BFA decoding system was discussed as well. It was shown that an optimal input buffer size can be found for a target decoding throughput under a fixed hardware area constraint.

ACKNOWLEDGMENT

The authors would like to thank the Telecommunications Research Laboratory (TRL) of Toshiba Research Europe Ltd. and its directors for supporting this work.

REFERENCES

- [1] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, 2nd ed. Upper Saddle River, NJ: Pearson Prentice-Hall, 2004.
- [2] F. Jelinek, "Fast sequential decoding using a stack," *IBM J. Res. Devel.*, vol. 13, pp. 675-685, Nov. 1969.
- [3] R. M. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Transactions on Information Theory*, vol. IT-9, no. 2, pp. 64-74, Apr. 1963.
- [4] R. O. Ozdag and P. A. Beerel, "An asynchronous low-power high-performance sequential decoder implemented with QDI templates," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 9, pp. 975-985, Sep. 2006.
- [5] M. Benaissa and Y. Zhu, "Reconfigurable hardware architectures for sequential and hybrid decoding," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 54, no. 3, pp. 555-565, Mar. 2007.
- [6] "Wireless High-Definition (WirelessHD)"; <http://www.wirelesshd.org/>
- [7] R. Xu, T. Kocak, G. Woodward, K. Morris and C. Dolwin, "Bidirectional Fano Algorithm for High Throughput Sequential Decoding," *IEEE Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Tokyo, Japan, 2009.
- [8] A. Martinez and M. Rovini, "Iterative decoders based on statistical multiplexing," *Proc. 3rd Int. Symp. on Turbo Codes and Related Topics*, pp. 423-426, Brest, France, 2003.
- [9] M. Rovini and A. Martinez, "On the Addition of an Input Buffer to an Iterative Decoder for LDPC Codes," *Proc. IEEE 65th Vehicular Technology Conference, VTC2007-Spring*, pp. 1995-1999, Apr. 2007.
- [10] S. L. Sweatlock, S. Dolinar, and K. Andrews, "Buffering Requirements for Variable Iterations LDPC Decoders," *Proc. Information Theory and Applications (ITA) Workshop*, pp. 523-530, 2008.
- [11] G. Bosco, G. Montorsi, and S. Benedetto, "Decreasing the Complexity of LDPC Iterative Decoders," *IEEE Communications Letters*, vol. 9, no. 7, pp. 634-636, July 2005.